A distributed algorithm for computing and updating the process number of a forest

David Coudert, Florian Huc, Dorian Mazauric

A distributed algorithm for computing and updating the process number of a forest*

(brief announcement)

David Coudert, Florian Huc, and Dorian Mazauric
MASCOTTE
INRIA, I3S, CNRS UMR6070, University of Nice Sophia
Sophia Antipolis, France

July 15, 2008

Treewidth and pathwidth have been introduced by Robertson and Seymour as part of the graph minor project. Those parameters are very important since many problems can be solved in polynomial time for graphs with bounded treewidth or pathwidth. By definition, the treewidth of a tree is one, but its pathwidth might be up to $\log n$. A linear time centralized algorithms to compute the pathwidth of a tree has been proposed in [1], but so far no dynamic algorithm exists.

The algorithmic counter part of the notion of pathwidth is the cops and robber game or node graph searching problem [2]. It consists in finding an invisible and fast fugitive in a graph using the smallest set of agents. A search strategy in a graph G can be defined as a serie of the following actions: (i) put an agent on a node, and (ii) remove an agent from a node if all its neighbors are either cleared or occupied by an agent. The node is now cleared. The fugitive is caught when all nodes are cleared. The minimum number of agents needed to clear the graph is the node search number (ns), and gives the pathwidth (pw) [3]. More precisely, it has been proved that $\mathrm{ns}(G) = \mathrm{pw}(G) + 1$ [4].

Other graph invariants closely related to the notion of pathwidth have been proposed (see [5] for a recent survey) such as the process number (pn) [6] and the edge search number (es), but so far it is not known if those parameters are strictly equivalent to pathwidth. However, we know that $\mathrm{pw}(G) \leq \mathrm{pn}(G) \leq \mathrm{pw}(G) + 1$ [6] and $\mathrm{pw}(G) \leq \mathrm{es}(G) \leq \mathrm{pw}(G) + 2$ [2]. A process strategy can be defined similary to a search strategy with the extra rule that the fugitive is forced to move at each round. Therefore, a node is also cleared when all its neighbors are occupied by an agent (the node is surrounded). For examples, $\mathrm{pn}(K_n) = n-1 = \mathrm{ns}(K_n)-1$, where K_n is a n-clique, and $\mathrm{pn}(C_k) = \mathrm{ns}(C_k) = 3$,

*This work has been supported by the European projects IST FET AEOLUS and COST 293 GRAAL, ARC CARMA, ANR JC OSERA, CRC CORSO, and Région PACA.

where C_k is a cycle of length $k \geq 5$. The process number is the minimum number of agents needed.

Here we propose a distributed algorithm to compute those parameters on trees and to update them in a forest after the addition or deletion of any edge [7]. Only initial conditions differ from one parameter to another. It is fully distributed, can be executed in an asynchronous environment and needs the transmission of only a small amount of information. It uses Theorem 1 which is also true for other parameters [6] and enforces each parameter to grow by 1, thus implying that for any tree $\mathrm{ns}(T)$, $\mathrm{es}(T)$, $\mathrm{pw}(T)$, and $\mathrm{pn}(T)$ are less than $\log_3(n)$.

Theorem 1 ([8]) *Let G_i, $i = 1, 2, 3$ be such that $\mathrm{ns}(G_i) = k > 1$. The graph G obtained by connecting each of the G_i's to a new node v is such that $\mathrm{ns}(G) = k + 1$.*

The principle of our algorithm, `algoHD`, is to perform a hierarchical decomposition of the tree. Each node collects a compact view of the subtree rooted at each of its soons, computes a compact view of the subtree it forms and sends it to its father, thus constructing a the hierarchical decomposition. A similar idea was used in [4] to design an algorithm computing the node search number in linear time. However their algorithm is centralized and its distributed version will transmit $\log \log n$ times more bits than ours. So we obtained,

Lemma 1 *Given a n nodes tree T, `algoHD` computes $\mathrm{pn}(T)$, $\mathrm{ns}(T)$ or $\mathrm{es}(T)$, in n steps, overall $O(n \log n)$ operations, and $n - 1$ messages of $\log_3 n + 2$ bits.*

We have extended our algorithm to a fully dynamic algorithm, `IncHD`, allowing to add or remove any edge. Each update can be performed in $O(D)$ steps, each of time complexity $O(\log n)$, and using $O(D)$ messages of $\log_3 n + 3$ bits, where D is the diameter of the tree. We have also extended our algorithms to trees and forests of unknown sizes by using messages of size $2L(t) + 4 + \varepsilon$, where $\varepsilon = 1$ for `IncHD`, and $L(t) \leq \mathrm{pn}(T) \leq \log_3 n$ is the minimum number of bits required to encode the local view of a subtree.

Finally, we have characterized the trees for which the process number (resp. edge search number) equals the node search number and so the pathwidth.

Lemma 2 *Given a tree T, $\mathrm{pn}(T) = \mathrm{pw}(T) + 1 = p + 1$ (resp. $\mathrm{pn}(T) = \mathrm{es}(T) + 1 = p+1$) iff there is a node v such that any components of $T - \{v\}$ has pathwidth at most p and there is at least three components with process number (resp. edge search number) p of which at most two have pathwidth p.*

A challenging task is now to give such characterisations for more general classes of graphs, as well as distributed and dynamic algorithms.

References

[1] K. Skodinis. Construction of linear tree-layouts which are optimal with respect to vertex separation in linear time. *Journal of Algorithms*, 47(1):40–59, 2003.

[2] M. Kirousis and C.H. Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(2):205–218, 1986.

[3] N. G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42(6):345–350, 1992.

[4] J.A. Ellis, I.H. Sudborough, and J.S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1):50–79, 1994.

[5] F. V. Fomin and D. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science, Special Issue on Graph Searching*, to appear.

[6] D. Coudert, S. Perennes, Q.-C. Pham, and J.-S. Sereni. Rerouting requests in wdm networks. In *AlgoTel'05*, pages 17–20, Presqu'île de Giens, France, mai 2005.

[7] D. Coudert, F. Huc, and D. Mazauric. A distributed algorithm for computing and updating the process number of a forest. Research Report 6560, INRIA, 06 2008.

[8] T. D. Parsons. Pursuit-evasion in a graph. In *Theory and applications of graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. Springer, Berlin, 1978.